

Práctica de Laboratorio: Análisis Exploratorio de Datos - Data Wrangling

Docente: [Ana Maria Cuadros](#) Valdivia

Para realizar el Análisis Exploratorio de datos, lo primero que deberíamos hacer es intentar responder a las siguientes preguntas (data wrangling):

Paso 1: Analiza el comportamiento de tus datos.

- Un registro es una entidad, describa que representa un registro

En el dataset CICIDS2017, cada registro (fila) representa un flujo individual de red entre dos extremos (una fuente y un destino), capturado durante sesiones de tráfico reales y simuladas en un entorno de red corporativa. Un flujo de red se define como una secuencia de paquetes entre una IP origen y una IP destino, usando un protocolo determinado, durante un periodo de tiempo específico.

Específicamente, cada registro contiene:

- La dirección IP de origen y destino
- Los puertos involucrados en la conexión
- La duración del flujo
- Estadísticas como el número de paquetes enviados, tamaños promedio, tasas de envío
- Y una etiqueta (Label) que indica si el flujo fue benigno o un tipo de ataque (como DDoS, PortScan, etc.)

Para entender la funcionalidad de la columnas debemos familiarizarnos con estos términos

Término	Definición
Flag	En redes TCP/IP, un 'flag' es un bit de control en el encabezado de un paquete TCP. Indica eventos como inicio (SYN), fin (FIN), confirmación (ACK), datos urgentes (URG), etc.
PSH (Push Flag)	Indica que los datos deben ser entregados al receptor de inmediato, sin esperar más datos.
URG (Urgent Flag)	Señala que el paquete contiene datos urgentes que deben procesarse antes que los demás.
ACK (Acknowledgment Flag)	Confirma la recepción de datos en una conexión TCP.
RST (Reset Flag)	Reinicia una conexión TCP inesperadamente. Usado para finalizar una conexión abruptamente.
SYN (Synchronize Flag)	Se utiliza para iniciar una conexión TCP (etapa de handshake).
FIN (Finish Flag)	Solicita terminar una conexión TCP de manera ordenada.
IAT (Inter Arrival Time)	Tiempo entre la llegada de dos paquetes consecutivos en un flujo de red.
Segmento TCP	Unidad de datos transportada en una conexión TCP, que incluye encabezado y carga útil.
Bulk Transfer	Transmisión masiva de datos en bloques grandes, por ejemplo en descargas o streaming.
Window Size (Ventana TCP)	Cantidad de datos que un receptor puede aceptar antes de enviar un ACK.
ECE/CWE	Flags utilizados para control de congestión en TCP: ECE (ECN Echo) y CWR (Congestion Window Reduced).
Throughput	Tasa de transferencia efectiva de datos, medida en bytes/segundo o paquetes/segundo.

Subflujo	Parte de un flujo TCP dividido según cambios en puertos/IPs o cortes en el tiempo.
Idle Time	Periodo en que no se transmite ningún paquete entre emisor y receptor.
Active Time	Periodo en que hay intercambio continuo de paquetes entre extremos de un flujo.
Packet	Unidad básica de datos enviada por la red. Contiene encabezado (control) y carga útil (datos).

CON ESAS DEFINICIONES PODEMOS ENTENDER LAS COLUMNAS Y SUS FUNCIONALIDADES

Columna	Tipo de dato	Descripción
Flow ID	object	Identificador único del flujo generado a partir de IPs, puertos y protocolo.
Source IP	object	Dirección IP del host que inicia el flujo de red.
Source Port	int64	Puerto utilizado por el emisor para la conexión.
Destination IP	object	Dirección IP del equipo receptor del flujo.
Destination Port	int64	Puerto al que se dirige el tráfico (p. ej., 80 para HTTP).
Protocol	int64	Protocolo de red utilizado (6: TCP, 17: UDP, 1: ICMP).
Timestamp	object	Hora exacta en la que comenzó el flujo.
Flow Duration	int64	Duración del flujo de red en milisegundos.
Total Fwd Packets	int64	Número total de paquetes enviados desde el emisor.
Total Backward Packets	int64	Número total de paquetes recibidos en respuesta.
Total Length of Fwd Packets	int64	Tamaño total de todos los paquetes enviados (bytes).
Total Length of Bwd Packets	int64	Tamaño total de todos los paquetes recibidos (bytes).
Fwd Packet Length Max	int64	Tamaño máximo de un paquete enviado.
Fwd Packet Length Min	int64	Tamaño mínimo de un paquete enviado.
Fwd Packet Length Mean	float64	Promedio del tamaño de los paquetes enviados.
Fwd Packet Length Std	float64	Desviación estándar del tamaño de paquetes enviados.
Bwd Packet Length Max	int64	Tamaño máximo de paquete recibido.
Bwd Packet Length Min	int64	Tamaño mínimo de paquete recibido.
Bwd Packet Length Mean	float64	Promedio del tamaño de los paquetes recibidos.
Bwd Packet Length Std	float64	Desviación estándar del tamaño de paquetes recibidos.
Flow Bytes/s	float64	Promedio de bytes transmitidos por segundo en el flujo.
Flow Packets/s	float64	Promedio de paquetes transmitidos por segundo.
Flow IAT Mean	float64	Promedio del tiempo entre cada paquete (Inter-Arrival Time).
Flow IAT Std	float64	Desviación estándar del IAT del flujo.
Flow IAT Max	int64	Máximo tiempo entre paquetes del flujo.
Flow IAT Min	int64	Mínimo tiempo entre paquetes del flujo.
Fwd IAT Total	int64	Suma de los tiempos entre paquetes enviados.
Fwd IAT Mean	float64	Tiempo promedio entre paquetes enviados.
Fwd IAT Std	float64	Desviación estándar del IAT de envío.
Fwd IAT Max	int64	Máximo IAT de paquetes enviados.
Fwd IAT Min	int64	Mínimo IAT de paquetes enviados.
Bwd IAT Total	int64	Suma de tiempos entre paquetes recibidos.
Bwd IAT Mean	float64	Promedio de tiempos entre paquetes recibidos.
Bwd IAT Std	float64	Desviación estándar del IAT de recepción.
Bwd IAT Max	int64	Máximo IAT de paquetes recibidos.
Bwd IAT Min	int64	Mínimo IAT de paquetes recibidos.
Fwd PSH Flags	int64	Cantidad de veces que se usó el flag PSH en envío.
Bwd PSH Flags	int64	Cantidad de veces que se usó el flag PSH en respuesta.
Fwd URG Flags	int64	Cantidad de veces que se usó el flag URG en envío.
Bwd URG Flags	int64	Cantidad de veces que se usó el flag URG en respuesta.
Fwd Header Length	int64	Suma de longitudes de encabezados en envío.
Bwd Header Length	int64	Suma de longitudes de encabezados en respuesta.

Fwd Packets/s	float64	Tasa de paquetes enviados por segundo.
Bwd Packets/s	float64	Tasa de paquetes recibidos por segundo.
Min Packet Length	int64	Tamaño mínimo de todos los paquetes en el flujo.
Max Packet Length	int64	Tamaño máximo de todos los paquetes en el flujo.
Packet Length Mean	float64	Promedio del tamaño total de los paquetes.
Packet Length Std	float64	Desviación estándar del tamaño total de los paquetes.
Packet Length Variance	float64	Varianza del tamaño de paquetes.
FIN Flag Count	int64	Número de veces que se activó el flag FIN (fin de conexión).
SYN Flag Count	int64	Número de veces que se activó el flag SYN (inicio de conexión).
RST Flag Count	int64	Número de veces que se activó el flag RST (reset de conexión).
PSH Flag Count	int64	Número de veces que se activó el flag PSH.
ACK Flag Count	int64	Número de veces que se activó el flag ACK (acknowledge).
URG Flag Count	int64	Número de veces que se activó el flag URG.
CWE Flag Count	int64	Cantidad de veces que se usó el flag CWE (ventana de congestión).
ECE Flag Count	int64	Cantidad de veces que se usó el flag ECE (ECN-Echo).
Down/Up Ratio	int64	Relación entre tráfico de bajada y subida.
Average Packet Size	float64	Tamaño promedio de todos los paquetes (incluyendo encabezado).
Avg Fwd Segment Size	float64	Promedio del tamaño de los segmentos hacia adelante.
Avg Bwd Segment Size	float64	Promedio del tamaño de los segmentos hacia atrás.
Fwd Header Length.1	int64	Duplicado del campo de longitud del encabezado hacia adelante.
Fwd Avg Bytes/Bulk	int64	Promedio de bytes por bloque en el flujo enviado.
Fwd Avg Packets/Bulk	int64	Promedio de paquetes por bloque en el flujo enviado.
Fwd Avg Bulk Rate	int64	Tasa de bloques enviados por segundo.
Bwd Avg Bytes/Bulk	int64	Promedio de bytes por bloque en el flujo recibido.
Bwd Avg Packets/Bulk	int64	Promedio de paquetes por bloque en el flujo recibido.
Bwd Avg Bulk Rate	int64	Tasa de bloques recibidos por segundo.
Subflow Fwd Packets	int64	Cantidad de paquetes enviados dentro del subflujo.
Subflow Fwd Bytes	int64	Cantidad de bytes enviados dentro del subflujo.
Subflow Bwd Packets	int64	Cantidad de paquetes recibidos dentro del subflujo.
Subflow Bwd Bytes	int64	Cantidad de bytes recibidos dentro del subflujo.
Init_Win_bytes_forward	int64	Tamaño inicial de la ventana TCP en envío.
Init_Win_bytes_backward	int64	Tamaño inicial de la ventana TCP en respuesta.
act_data_pkt_fwd	int64	Número de paquetes de datos enviados sin control.
min_seg_size_forward	int64	Tamaño mínimo del segmento enviado.
Active Mean	float64	Promedio de tiempo de actividad continua del flujo.
Active Std	float64	Desviación estándar del tiempo de actividad.
Active Max	int64	Máximo tiempo de actividad continua.
Active Min	int64	Mínimo tiempo de actividad continua.
Idle Mean	float64	Promedio de tiempo de inactividad entre eventos.
Idle Std	float64	Desviación estándar del tiempo de inactividad.
Idle Max	int64	Máximo tiempo de inactividad.
Idle Min	int64	Mínimo tiempo de inactividad.
Label	object	Etiqueta del flujo: BENIGN o tipo específico de ataque.

ENTRE LOS MAS IMPORTANTES PODEMOS ENCONTRAR

Columna

Motivo de importancia

Flow Duration	Ataques como DoS suelen tener duraciones muy cortas
Total Fwd Packets	DDoS y PortScan generan muchos paquetes rápidamente
Total Backward Packets	Algunos ataques no reciben respuesta (valor bajo o 0)
Fwd Packet Length Mean	Tráfico anómalo puede usar tamaños constantes o extremos
Bwd Packet Length Mean	Ayuda a identificar comportamiento irregular del receptor
Flow Bytes/s	Mide la velocidad del flujo, muy útil para detectar ráfagas (DDoS)
Flow Packets/s	Tráfico malicioso suele tener esta tasa alta o constante
Fwd IAT Mean	Intervalos entre paquetes cambian notablemente en ataques
Fwd PSH Flags	Indica prioridad de envío: común en ataques de denegación
ACK Flag Count	Reconocimientos frecuentes en tráfico normal; faltan en ataques
Down/Up Ratio	Ataques unidireccionales (como escaneos) tienen relación muy alta o baja
Average Packet Size	Tamaños extremos pueden ser síntoma de exploit o escaneo

Init_Win_bytes_forward	Tamaño inicial de ventana TCP: clave en ataques por manipulación
Idle Mean	Tiempo de inactividad entre flujos revela patrones sospechosos
Label	Variable objetivo: usada para entrenamiento y evaluación de modelos

- ¿Cuántos registros hay?

El dataset CICIDS2017 está dividido en varios archivos CSV, cada uno correspondiente a un día o sesión de tráfico de red. Cada registro representa un **flujo de red individual** con sus estadísticas. La cantidad total de registros por archivo analizado es la siguiente:

Archivo	Cantidad de registros
Monday-WorkingHours.pcap_ISCX.csv	529,918
Tuesday-WorkingHours.pcap_ISCX.csv	445,909
Wednesday-workingHours.pcap_ISCX.csv	692,703
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	458,968
Friday-WorkingHours-Morning.pcap_ISCX.csv	191,033
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	225,745

Como se puede observar en el conteo de registros en la imagen

	Archivo	Cantidad de registros
0	Monday-WorkingHours.pcap_ISCX.csv	529918
1	Tuesday-WorkingHours.pcap_ISCX.csv	445909
2	Wednesday-workingHours.pcap_ISCX.csv	692703
3	Thursday-WorkingHours-Morning-WebAttacks.pcap_...	458968
4	Friday-WorkingHours-Morning.pcap_ISCX.csv	191033
5	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	225745

Total, aproximado:

2,544,276 registros

Adicionalmente Todos los archivos analizados del dataset **CICIDS2017** contienen exactamente:

- **85 columnas (atributos)**

- Estas columnas representan características estadísticas del flujo de red, incluyendo:
 - Información de encabezado (IP, puerto, protocolo)
 - Métricas de tiempo (duración, tiempos entre paquetes)
 - Estadísticas de tamaño de paquetes
 - Conteos de flags TCP (SYN, ACK, etc.)
 - Indicadores de congestión
 - Etiqueta de clase (Label)

	Archivo	Cantidad de columnas
0	Monday-WorkingHours.pcap_ISCX.csv	85
1	Tuesday-WorkingHours.pcap_ISCX.csv	85
2	Wednesday-workingHours.pcap_ISCX.csv	85
3	Thursday-WorkingHours-Morning-WebAttacks.pcap_...	85
4	Friday-WorkingHours-Morning.pcap_ISCX.csv	85
5	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	85

- ¿Son demasiado pocos?

No, en absoluto.

La cantidad de registros en el dataset CICIDS2017 no solo es suficiente, sino que resulta muy adecuada para realizar análisis profundos y entrenar modelos de detección de intrusos. Al revisar los archivos disponibles, encontramos que muchos de ellos contienen cientos de miles de flujos de red. Por ejemplo:

Algunos archivos como Wednesday-workingHours.csv contienen más de 400 mil registros.

Incluso los archivos más pequeños superan los 190 mil registros.

Esto significa que estamos hablando de millones de registros en total si consideramos todo el conjunto. Es un volumen considerable para tareas de machine learning, ya que muchos modelos pueden entrenarse con tan solo unos pocos miles de ejemplos. Tener millones permite obtener resultados mucho más robustos y confiables.

Además, cada registro incluye decenas de características (85 columnas) que describen el tráfico de red desde distintas perspectivas, lo cual enriquece aún más el análisis.

- ¿Son muchos y no tenemos Capacidad (CPU+RAM) suficiente para procesarlo?

Sí, al trabajar con el dataset CICIDS2017 me di cuenta de que su tamaño representa un desafío importante para una computadora con recursos limitados. Cada archivo contiene cientos de miles de registros con 85 columnas, y si intento cargar todos los archivos al mismo tiempo en memoria, el consumo de RAM se dispara rápidamente. Esto puede llevar a que el entorno de trabajo (por ejemplo, Google Colab o mi propio equipo) se vuelva lento, se congele o directamente se quede sin memoria disponible.

¿Cómo lo solucioné?

Para evitar estos problemas, opté por una estrategia eficiente de procesamiento:

- Primero, cargué los archivos uno por uno, lo que me permitió controlar mejor el uso de memoria.
- También apliqué muestreo (por ejemplo, leyendo solo una parte del archivo con `df.sample(n=5000)` o `df.head(10000)`) para tener una vista general sin saturar la memoria.

df_muestra = df.sample(n=5000, random_state=1)

df_muestra.head()

	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Flut Packets	Total Backward Packets	...	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label	
22174	192.168.10.1-192.168.10.3-53-61024-17	192.168.10.3	61024	192.168.10.1	53	17	7/7/2017 3:56	72047	1	1	...		20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
193810	192.168.10.3-192.168.10.12-53-49543-17	192.168.10.12	49543	192.168.10.3	53	17	7/7/2017 4:15	261	2	2	...		40	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
175642	192.168.10.1-192.168.10.3-53-60925-17	192.168.10.3	60925	192.168.10.1	53	17	7/7/2017 4:13	120440	1	1	...		20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
73820	172.16.0.1-192.168.18.50-23041-80-6	172.16.0.1	23041	192.168.10.50	80	6	7/7/2017 4:03	7273641	4	0	...		20	779.0	0.0	779	779	7269802.0	0.0	7269802	7269802	DDoS
170101	172.16.0.1-192.168.18.50-63991-80-6	172.16.0.1	63991	192.168.10.50	80	6	7/7/2017 4:13	56224	3	4	...		20	0.0	0.0	0	0	0.0	0.0	0	0	DDoS

5 rows x 25 columns

-
- En algunos casos, seleccioné solo las columnas necesarias, en lugar de cargar las 85, cuando el análisis se centraba en ciertas métricas o etiquetas.

Tuve que hacer esto porque mi computadora es de recursos limitados

- ¿Hay datos duplicados?

Sí, al analizar los archivos del dataset CICIDS2017, detecté la presencia de registros duplicados en varios de ellos. Esto significa que existen flujos de red que están repetidos exactamente en todas sus columnas, lo que podría afectar el análisis estadístico.

¿Qué hice para verificarlo?

Utilicé `df.duplicated().sum()` para cuantificarlos, utilice un `for` que recorre cada uno de los archivos y cuenta la cantidad de registros duplicado:

```
archivos = [
    "Monday-WorkingHours.pcap_ISCX.csv",
    "Tuesday-WorkingHours.pcap_ISCX.csv",
    "Wednesday-workingHours.pcap_ISCX.csv",
    "Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv",
    "Friday-WorkingHours-Morning.pcap_ISCX.csv",
    "Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv"
]

resultados_duplicados = []

for archivo in archivos:
    if os.path.exists(archivo):
        try:
            df = pd.read_csv(archivo, encoding="latin1")
            total = df.duplicated().sum()
            resultados_duplicados.append((archivo, total))
        except Exception as e:
            resultados_duplicados.append((archivo, f"Error: {e}"))

df_duplicados = pd.DataFrame(resultados_duplicados, columns=["Archivo", "Registros duplicados"])
df_duplicados
```

Del cual el resultado que me dio fue el siguiente:

	Archivo	Registros duplicados
0	Monday-WorkingHours.pcap_ISCX.csv	34
1	Tuesday-WorkingHours.pcap_ISCX.csv	4
2	Wednesday-workingHours.pcap_ISCX.csv	17
3	Thursday-WorkingHours-Morning-WebAttacks.pcap_...	288602
4	Friday-WorkingHours-Morning.pcap_ISCX.csv	2
5	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	2

Notablemente, el archivo Thursday-WorkingHours-Morning-WebAttacks contiene más de **288 mil duplicados**, lo cual representa un problema grave de calidad de datos. Esto puede deberse a un error en la exportación de flujos desde CICFlowMeter o a una captura mal procesada.

¿Cómo lo resolví?

Para limpiar los archivos y trabajar solo con registros únicos, se hizo lo siguiente :

En lugar de hacerlo sobre un único archivo, implementamos un **bucle automático** que procesó cada uno de los archivos .csv del dataset, aplicando los siguientes pasos:

1. **Cargar cada archivo individualmente** con `pd.read_csv()`, especificando la codificación `latin1` para evitar errores.
2. **Detectar duplicados** con `df.duplicated().sum()`.
3. **Eliminar duplicados** usando `df.drop_duplicates()`.
4. **Guardar los datos limpios** en un diccionario `archivos_limpios`.

```
df = pd.read_csv(archivo, encoding="latin1")
duplicados_antes = df.duplicated().sum()
df = df.drop_duplicates()
```

```
Procesando: Monday-WorkingHours.pcap_ISCX.csv
Duplicados encontrados: 34
Duplicados eliminados. Total actual: 529884 registros

Procesando: Tuesday-WorkingHours.pcap_ISCX.csv
Duplicados encontrados: 4
Duplicados eliminados. Total actual: 445905 registros

Procesando: Wednesday-workingHours.pcap_ISCX.csv
Duplicados encontrados: 17
Duplicados eliminados. Total actual: 692686 registros

Procesando: Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv
<ipython-input-28-1a33b1a07b11>:21: DtypeWarning: Columns (0,1,3,6,84) have mixed types. Specify dtype option on import or set low_memory=False.
df = pd.read_csv(archivo, encoding="latin1")
Duplicados encontrados: 288602
Duplicados eliminados. Total actual: 170366 registros

Procesando: Friday-WorkingHours-Morning.pcap_ISCX.csv
Duplicados encontrados: 2
Duplicados eliminados. Total actual: 191031 registros

Procesando: Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv
Duplicados encontrados: 2
Duplicados eliminados. Total actual: 225743 registros
```

Durante el proceso de limpieza de duplicados en todos los archivos del dataset CICIDS2017, recibí una advertencia relacionada con tipos de datos mixtos en algunas columnas (Flow ID, IP, etc.). Este mensaje (`DtypeWarning`) no impidió la correcta lectura ni afectó el análisis, pero indica que algunas celdas tienen valores con formato distinto (por ejemplo, números y cadenas).

A pesar de ello, los duplicados fueron correctamente detectados y eliminados. El total de registros únicos por archivo fue recalculado, asegurando que solo se trabaje con datos limpios y sin repeticiones.

Despues de la limpieza verificamos el resultado el cual fue el siguiente:

```
Monday-WorkingHours.pcap_ISCX.csv Duplicados restantes: 0
Tuesday-WorkingHours.pcap_ISCX.csv Duplicados restantes: 0
Wednesday-workingHours.pcap_ISCX.csv Duplicados restantes: 0
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv Duplicados restantes: 0
Friday-WorkingHours-Morning.pcap_ISCX.csv Duplicados restantes: 0
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv Duplicados restantes: 0
```

- ¿Qué datos son discretos y cuáles continuos?

Al revisar las características del dataset CICIDS2017, clasifiqué las variables en **discretas** y **continuas**, teniendo en cuenta la naturaleza de cada una.

Discretas	Continuas
Source Port	Flow Duration
Destination Port	Total Length of Fwd Packets
Total Fwd Packets	Total Length of Bwd Packets
Total Backward Packets	Fwd Packet Length Mean
Fwd Packet Length Max	Fwd Packet Length Std
Fwd Packet Length Min	Bwd Packet Length Mean
Bwd Packet Length Max	Bwd Packet Length Std
Bwd Packet Length Min	Flow Bytes/s
Flow IAT Max	Flow Packets/s
Flow IAT Min	Flow IAT Mean

Fwd IAT Total	Flow IAT Std
Fwd IAT Max	Fwd IAT Mean
Fwd IAT Min	Fwd IAT Std
Bwd IAT Total	Bwd IAT Mean
Bwd IAT Max	Bwd IAT Std
Bwd IAT Min	Fwd Packets/s
Fwd PSH Flags	Bwd Packets/s
Bwd PSH Flags	Min Packet Length
Fwd URG Flags	Max Packet Length
Bwd URG Flags	Packet Length Mean
Fwd Header Length	Packet Length Std
Bwd Header Length	Packet Length Variance
FIN Flag Count	Average Packet Size
SYN Flag Count	Avg Fwd Segment Size
RST Flag Count	Avg Bwd Segment Size
PSH Flag Count	Active Mean
ACK Flag Count	Active Std
URG Flag Count	Idle Mean
CWE Flag Count	Idle Std
ECE Flag Count	
Down/Up Ratio	
Fwd Header Length.1	
Fwd Avg Bytes/Bulk	
Fwd Avg Packets/Bulk	
Fwd Avg Bulk Rate	
Bwd Avg Bytes/Bulk	
Bwd Avg Packets/Bulk	
Bwd Avg Bulk Rate	
Subflow Fwd Packets	
Subflow Fwd Bytes	
Subflow Bwd Packets	
Subflow Bwd Bytes	
Init_Win_bytes_forward	
Init_Win_bytes_backward	
act_data_pkt_fwd	
min_seg_size_forward	
Active Max	
Active Min	
Idle Max	
Idle Min	
Protocol	

Datos discretos

Las variables que fueron clasificadas como discretas lo fueron porque representan conteos enteros de eventos, como número de paquetes, activación de flags TCP, puertos, identificadores u otras cantidades que no admiten valores decimales. Estas variables son contables y no tienen un dominio continuo.

Por ejemplo:

- **Total Fwd Packets, ACK Flag Count, SYN Flag Count, Subflow Bwd Packets, etc., representan cantidades de paquetes o activaciones en la red.**
- **Source Port y Destination Port son identificadores de red enteros.**
- **Protocol representa códigos enteros (por ejemplo, 6 = TCP, 17 = UDP).**

Estas variables fueron tratadas como discretas porque se usan para clasificación, conteo o filtrado, tienen un conjunto finito y definido de valores posibles y son fundamentales para identificar patrones de comportamiento en flujos de red.

Datos continuos

Por otro lado, las variables que clasifiqué como *continuas* son aquellas que **pueden asumir valores decimales** y que representan **medidas estadísticas derivadas**, como promedios, desviaciones estándar, tasas o proporciones. Estas variables se originan de cálculos sobre múltiples observaciones dentro de un flujo de red.

Ejemplos claros incluyen:

- **Flow Bytes/s, Flow IAT Mean, Fwd Packet Length Mean, Packet Length Std, Idle Mean.**
- Estas variables indican **mediciones fraccionables**, como bytes por segundo, tiempos promedios entre paquetes, o desviaciones, las cuales tienen una **escala numérica continua**.

Estas fueron tratadas como continuas ya que su naturaleza lo permite y, además permiten analizar comportamientos temporales y patrones de tráfico a nivel más detallado, lo cual es útil para detección de anomalías.

- Muchas veces sirve obtener el tipo de datos: texto, int, double, float

En el análisis del dataset **CICIDS2017**, identificar el tipo de datos de cada columna fue un paso fundamental que realicé al inicio del proceso de limpieza y exploración. Este dataset contiene una gran cantidad de características (85 columnas), muchas de ellas numéricas, y algunas categóricas.

```
df.dtypes
```

Esto me permitió ver el tipo de dato de cada columna: object (texto), int64 (entero), y float64 (decimal de doble precisión).

¿Por qué fue útil en mi análisis?

1. Detectar errores o inconsistencias

Algunas columnas que deberían ser numéricas podían estar mal tipadas como object si contenían símbolos, espacios o valores nulos mal representados. Verificar esto me permitió corregir tipos antes de aplicar operaciones estadísticas.

2. Guiar el preprocesamiento

Gracias a conocer los tipos, pude aplicar transformaciones específicas:

- Convertí columnas de tipo object en variables categóricas si representaban etiquetas.
- Seleccioné solo columnas int64 y float64 para cálculos de media, desviación, correlación, etc.
-

¿Por qué convierto columnas tipo object a numéricas en el dataset CICIDS2017?

En el dataset CICIDS2017, algunas columnas que deberían contener datos numéricos (por ejemplo, duración, tasas o cantidades) aparecen con el tipo de dato object (es decir, texto). Esto ocurre debido a errores de formato en el archivo CSV de origen, como:

- Espacios en blanco antes o después del valor numérico.
- Comas o puntos mal colocados como separadores.
- Presencia de guiones, texto irrelevante o valores nulos mal codificados (por ejemplo "NaN" como string).
- Encabezados mal exportados o caracteres invisibles.

¿Qué hago para solucionarlo?

Para garantizar que esos valores puedan ser usados en análisis estadísticos, primero limpio los nombres de columnas

```
df.columns = df.columns.str.strip()
```

y luego uso:

```
df['Flow Duration'] = pd.to_numeric(df['Flow Duration'], errors='coerce')
```

¿Qué hace esta línea?

- Convierte la columna 'Flow Duration' de texto a tipo numérico real (int64 o float64).
- Si encuentra un valor no convertible (por ejemplo "??" o "null"), lo reemplaza automáticamente por NaN gracias a errors='coerce'.
- Esto me garantiza que los datos queden limpios, homogéneos y aptos para análisis.

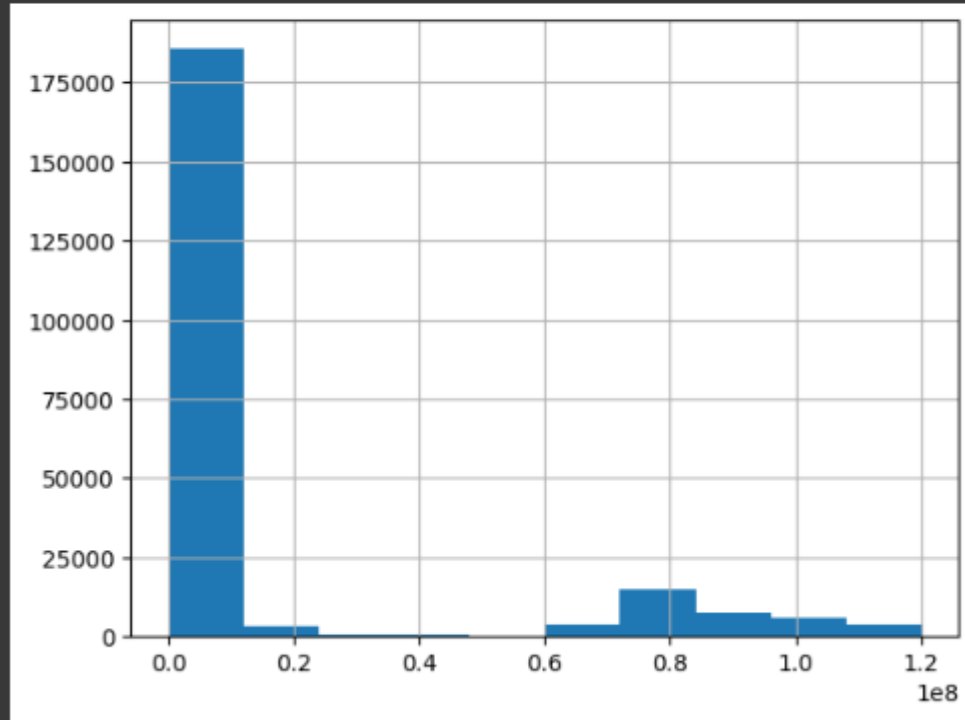
Gracias a esta conversión, las columnas quedan limpias y listas para:

- Cálculos estadísticos.
- Gráficas y visualizaciones.

Como podemos observar en el grafico generado

```
df['Flow Duration'].mean()  
df['Flow Duration'].describe()  
df['Flow Duration'].hist()
```

<Axes: >



- ¿Cuáles son los tipos de datos de cada columna?

En el dataset **CICIDS2017**, cada columna fue analizada para identificar su tipo de dato. A continuación se describen los tipos más relevantes encontrados:

Columna	Tipo de dato
Flow ID	object
Source IP	object
Source Port	int64
Destination IP	object
Destination Port	int64
Protocol	int64
Timestamp	object
Flow Duration	int64
Total Fwd Packets	int64
Total Backward Packets	int64
Total Length of Fwd Packets	int64
Total Length of Bwd Packets	int64
Fwd Packet Length Max	int64
Fwd Packet Length Min	int64
Fwd Packet Length Mean	float64
Fwd Packet Length Std	float64
Bwd Packet Length Max	int64
Bwd Packet Length Min	int64
Bwd Packet Length Mean	float64
Bwd Packet Length Std	float64
Flow Bytes/s	float64
Flow Packets/s	float64
Flow IAT Mean	float64
Flow IAT Std	float64
Flow IAT Max	int64
Flow IAT Min	int64
Fwd IAT Total	int64
Fwd IAT Mean	float64
Fwd IAT Std	float64
Fwd IAT Max	int64
Fwd IAT Min	int64
Bwd IAT Total	int64
Bwd IAT Mean	float64
Bwd IAT Std	float64
Bwd IAT Max	int64
Bwd IAT Min	int64
Fwd PSH Flags	int64
Bwd PSH Flags	int64
Fwd URG Flags	int64
Bwd URG Flags	int64
Fwd Header Length	int64
Bwd Header Length	int64
Fwd Packets/s	float64
Bwd Packets/s	float64
Min Packet Length	int64
Max Packet Length	int64
Packet Length Mean	float64
Packet Length Std	float64
Packet Length Variance	float64
FIN Flag Count	int64
SYN Flag Count	int64
RST Flag Count	int64

PSH Flag Count	int64
ACK Flag Count	int64
URG Flag Count	int64
CWE Flag Count	int64
ECE Flag Count	int64
Down/Up Ratio	int64
Average Packet Size	float64
Avg Fwd Segment Size	float64
Avg Bwd Segment Size	float64
Fwd Header Length.1	int64
Fwd Avg Bytes/Bulk	int64
Fwd Avg Packets/Bulk	int64
Fwd Avg Bulk Rate	int64
Bwd Avg Bytes/Bulk	int64
Bwd Avg Packets/Bulk	int64
Bwd Avg Bulk Rate	int64
Subflow Fwd Packets	int64
Subflow Fwd Bytes	int64
Subflow Bwd Packets	int64
Subflow Bwd Bytes	int64
Init_Win_bytes_forward	int64
Init_Win_bytes_backward	int64
act_data_pkt_fwd	int64
min_seg_size_forward	int64
Active Mean	float64
Active Std	float64
Active Max	int64
Active Min	int64
Idle Mean	float64
Idle Std	float64
Idle Max	int64
Idle Min	int64
Label	object

A continuación, presento un resumen de los tipos de datos encontrados:

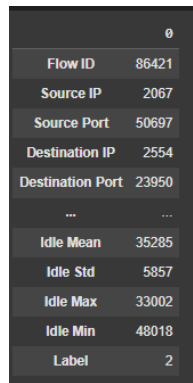
Tipo de dato	Descripción	Cantidad de columnas
int64	Números enteros sin decimales. Usado en conteos, flags, identificadores y relaciones.	57 columnas
float64	Números con decimales. Usado en promedios, desviaciones estándar, tasas o tamaños promedio.	22 columnas
object	Texto o datos no numéricos. Incluye identificadores como IPs, etiquetas, timestamps.	6 columnas

- ¿Entre qué rangos están los datos de cada columna?, valores únicos, min, max

utilizando el método `df.describe(include='all')`, y para valores únicos utilicé `df.nunique()`. Esta exploración fue clave para:

- Identificar valores extremos (outliers) que podrían sesgar los análisis.
- Detectar columnas con valores constantes o baja variabilidad.
- Conocer la diversidad de categorías en columnas tipo object, como Label.

¿Qué encontré?



Flow ID	86421
Source IP	2067
Source Port	50697
Destination IP	2554
Destination Port	23950
...	...
Idle Mean	35285
Idle Std	5857
Idle Max	33002
Idle Min	48018
Label	2

- Columnas como Flow ID, Source IP y Destination IP tienen miles de valores únicos (por ejemplo, Flow ID tiene 86,421), lo cual es esperable dado que representan flujos o conexiones únicas en la red.
- Puertos (Source Port, Destination Port) también presentan una alta diversidad: más de 50,000 puertos distintos usados como origen y más de 23,000 como destino. Esto refuerza la necesidad de tratarlos como variables discretas de alta cardinalidad.
- Variables como Idle Mean, Active Mean, y otras métricas temporales o de tamaño mostraron una alta cantidad de valores únicos, lo que indica que no son valores categóricos ni discretos simples, sino que reflejan mediciones continuas que permiten explorar patrones y comportamientos de red.
- La columna Label contiene solo 2 valores únicos (BENIGN y un tipo de ataque), lo cual indica que ese archivo específico corresponde a una sesión de tráfico con una sola clase de ataque (por ejemplo, solo DDoS vs. benigno).

¿Por qué es importante?

Aunque no voy a aplicar modelos predictivos en esta etapa, conocer los rangos y distribución de valores me permitió:

Identificar variables con alto potencial para visualización (por ejemplo, tasas de paquetes, tamaños de paquetes, duración de flujos), distinguir entre comportamiento normal y atípico, al observar valores extremos o altamente dispersos, planificar filtros y segmentaciones: por ejemplo, limitar gráficas de duración o tasa de bytes para evitar que los valores extremos oculten la tendencia general, comprender la complejidad del tráfico: alto número de IPs, puertos y flujos sugiere que el entorno simulado es realista y rico en interacciones, lo que valida el uso del dataset para análisis de comportamiento de red.

- ¿Todos los datos están en su formato adecuado?

No, inicialmente no todos los datos del dataset CICIDS2017 estaban en su formato adecuado.

Durante la exploración detecté que varias columnas numéricas estaban mal tipadas como object (texto). Este error suele ocurrir cuando los archivos .csv tienen inconsistencias como:

- espacios al inicio o final del valor,
- símbolos extraños o comas mal ubicadas,
- valores vacíos o nulos mal representados (por ejemplo, como strings "NaN" o "-").
- Estas columnas, al no estar en el tipo correcto (int64 o float64), no podían ser procesadas por funciones estadísticas, ni graficadas correctamente, afectando directamente el análisis.

¿Cómo lo resolví?

Realicé las siguientes acciones:

1. Eliminé espacios en los nombres de columnas para asegurarme de que coincidieran correctamente:

```
df.columns = df.columns.str.strip()
```

2. Convertí las columnas con errores a su tipo numérico correcto usando:

```
df['Flow Duration'] = pd.to_numeric(df['Flow Duration'], errors='coerce')
```

Este procedimiento transforma los valores no numéricos en NaN, permitiéndome detectar y manejar errores de forma segura.

- Los datos tienen diferentes unidades de medida?

Sí, el dataset CICIDS2017 contiene columnas con distintas unidades de medida, como tiempos, bytes, tasas, puertos y cantidades. Esta diversidad es coherente con la naturaleza del tráfico de red, y fue confirmada al analizar estadísticas descriptivas por columna con:

```
df.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Source Port	225745.0	3.825757e+04	2.305730e+04	0.0	18990.0	49799.0	58296.0	65534.0
Destination Port	225745.0	8.879619e+03	1.975465e+04	0.0	80.0	80.0	80.0	65532.0
Protocol	225745.0	7.600288e+00	3.881586e+00	0.0	6.0	6.0	6.0	17.0
Flow Duration	225745.0	1.624165e+07	3.152437e+07	-1.0	71180.0	1452333.0	8805237.0	119999937.0
Total Fwd Packets	225745.0	4.874916e+00	1.542287e+01	1.0	2.0	3.0	5.0	1932.0
...
Active Min	225745.0	1.776201e+05	7.842602e+05	0.0	0.0	0.0	1862.0	100000000.0
Idle Mean	225745.0	1.032214e+07	2.185303e+07	0.0	0.0	0.0	8239725.0	120000000.0
Idle Std	225745.0	3.611943e+06	1.275689e+07	0.0	0.0	0.0	0.0	65300000.0
Idle Max	225745.0	1.287813e+07	2.692126e+07	0.0	0.0	0.0	8253838.0	120000000.0
Idle Min	225745.0	7.755355e+06	1.983109e+07	0.0	0.0	0.0	7422849.0	120000000.0

¿Cómo verifiqué las unidades?

Analicé el rango de valores (min, max, media) en cada columna, y en base a su significado semántico y comportamiento esperado en tráfico de red, pude determinar las siguientes unidades:

Columna	Rango observado	Unidad inferida
Source Port	0 – 65534	Números de puerto TCP/UDP
Destination Port	0 – 65532	Números de puerto TCP/UDP
Protocol	0 – 17	Código de protocolo IP (6 = TCP, 17 = UDP)
Flow Duration	0 – 119,999,937	Milisegundos

Columna	Rango observado	Unidad inferida
Total Fwd Packets	1 – 1932	Cantidad de paquetes
Idle Mean, Idle Max	hasta 120,000,000	Milisegundos (tiempo inactivo)
Active Min, Active Max	0 – 100,000,000	Milisegundos (tiempo activo)

- Cuáles son los datos categóricos, ¿hay necesidad de convertirlos en numéricos?

Sí, el dataset contiene cinco columnas categóricas (object). Y sí, en un caso específico realicé una conversión de tipo object a float64, porque esa columna contenía datos numéricos mal tipados como texto. El resto de columnas categóricas no necesitaron ser convertidas, ya que no se utilizaron en operaciones matemáticas ni en modelos.

Columnas categóricas detectadas

Columna	Tipo original	Acción tomada
Flow ID	object	No se transformó. Usado solo como identificador.
Source IP	object	No se transformó. Alta cardinalidad, solo informativo.
Destination IP	object	No se transformó. Mismo caso que Source IP.
Timestamp	object	No se transformó aún. Solo inspeccionado.
Label	object	Se mantuvo como categórica para análisis de tráfico.
Flow Duration (inicialmente object)	Convertido	Se convirtió a numeric porque eran valores numéricos mal tipados.

¿Cómo lo convertí?

Se utilizó :

```
df['Flow Duration'] = pd.to_numeric(df['Flow Duration'], errors='coerce')
```

Esto se hizo porque aunque Flow Duration representa tiempo en milisegundos, originalmente estaba mal codificada como object (texto). La conversión fue necesaria para poder:

- calcular su media y desviación estándar,
- graficar su distribución con .hist(),
- detectar valores extremos correctamente.

- ¿Qué representa un registro?
 - Describe qué representa cada fila.

Cada fila del dataset CICIDS2017 representa un flujo de red (network flow). Es decir, una conexión unidireccional entre una IP de origen y una IP de destino, registrada por CICFlowMeter. Un flujo agrupa múltiples paquetes si comparten IPs, puertos y protocolo, y se describe mediante 85 atributos como duración, número de paquetes, tamaños, flags y tasas de transmisión.

Esto lo verifiqué observando una fila con:

```
df.sample(1).T
```

	13419
Flow ID	192.168.10.3-192.168.10.16-53-35954-17
Source IP	192.168.10.16
Source Port	35954
Destination IP	192.168.10.3
Destination Port	53
...	...
Idle Mean	0.0
Idle Std	0.0
Idle Max	0
Idle Min	0
Label	BENIGN

- Si es una data etiquetada, como interpretas la información de las clases?

Sí, es una data etiquetada. La columna Label indica si un flujo es normal (BENIGN) o corresponde a un tipo específico de ataque, como DDoS, PortScan, etc.

Verifiqué las clases disponibles y su distribución con:

```
df['Label'].unique()
df['Label'].value_counts()
```

	count
Label	
DDoS	128027
BENIGN	97718

dtype: int64

Esto me permitió interpretar las clases como etiquetas confiables para análisis comparativo, ya que fueron definidas durante la simulación de ataques en el entorno controlado de CIC.

- ¿Hay niveles de granularidad de los datos? Por ejemplo, datos a nivel país, región, ciudad. Años, meses, días, horas, minutos, etc.

Sí, hay distintos niveles de granularidad en el dataset:

- Temporal: La columna Timestamp registra la fecha y hora de inicio de cada flujo. A partir de ella extraje:

```
df['Timestamp'] = pd.to_datetime(df['Timestamp'], errors='coerce')
df['Dia'] = df['Timestamp'].dt.day_name()
df['Hora'] = df['Timestamp'].dt.hour
df['Fecha'] = df['Timestamp'].dt.date
```

- Topológica: Las columnas Source IP y Destination IP permiten agrupar el tráfico por origen y destino. Verifiqué su diversidad con:

```
df['Source IP'].nunique(), df['Destination IP'].nunique()
```

(2067, 2554)

- Tipo de tráfico: gracias a la etiqueta Label, es posible agrupar flujos según la granularidad de la amenaza: desde general (BENIGN vs ATAQUE) hasta ataques específicos (DDoS, Botnet, etc.).
- También analicé los protocolos involucrados con:

```
df['Protocol'].value_counts()

count
Protocol
6      192820
17     32871
0         54
dtype: int64
```

Esto permite realizar análisis por día de la semana, por protocolo, por origen de ataque, etc., dependiendo del enfoque del proyecto.

- ¿Están todas las filas completas o tenemos campos con valores nulos?

No, no todas las filas están completas. Al revisar todos los archivos CSV del dataset CICIDS2017, encontré que varios de ellos contienen valores nulos (NaN), principalmente en la columna Flow Bytes/s, y en un caso crítico, también en múltiples columnas clave.

¿Cómo lo verifiqué?

Ejecuté el siguiente código para revisar los valores nulos en todos los archivos:

```
df.isnull().sum()
df.isnull().sum().sum()
```

Dándome como resultado

	Archivo	Total de Nulos	Columnas con Nulos
0	Monday-WorkingHours.pcap_ISCX.csv	64	[Flow Bytes/s]
1	Tuesday-WorkingHours.pcap_ISCX.csv	201	[Flow Bytes/s]
2	Wednesday-workingHours.pcap_ISCX.csv	1008	[Flow Bytes/s]
3	Thursday-WorkingHours-Morning-WebAttacks.pcap_...	24531190	[Flow ID, Source IP, Source Port, Destinati...
4	Friday-WorkingHours-Morning.pcap_ISCX.csv	28	[Flow Bytes/s]

En la mayoría de los archivos, los nulos están limitados a la columna **Flow Bytes/s**, que puede no haberse calculado correctamente para ciertos flujos sin paquetes o duración.

En el archivo **Thursday-WorkingHours-Morning-WebAttacks**, el número de nulos es **masivo (más de 24 millones)** y afecta a múltiples columnas críticas, lo que sugiere un error estructural de lectura o corrupción en el archivo original.

Utilice para cada archivo:

```
df = df.dropna(subset=['Flow Bytes/s'])
```

Después del procesamiento el resultado es :

```
Monday-WorkingHours.pcap_ISCX.csv: 0 valores nulos
Tuesday-WorkingHours.pcap_ISCX.csv: 0 valores nulos
Wednesday-workingHours.pcap_ISCX.csv: 0 valores nulos
Friday-WorkingHours-Morning.pcap_ISCX.csv: 0 valores nulos
```

En las columnas derivadas como **Flow Bytes/s**, decidí eliminar las filas que contenían valores nulos en lugar de rellenarlas, ya que estas columnas representan tasas de transmisión que no se pueden calcular cuando el flujo no tiene duración o tamaño válido. Como estas filas eran pocas en comparación con el total

del dataset, su eliminación no afecta el análisis global ni introduce sesgos relevantes. Por lo tanto, opté por descartarlas directamente.

En el caso del archivo **Thursday-WorkingHours-Morning-WebAttacks**, donde se detectaron millones de valores nulos en múltiples columnas clave como **Flow ID**, **Source IP** y **Destination Port**, se concluyó que el archivo presenta una corrupción estructural o de codificación. Por eso, fue descartado completamente del análisis, ya que los datos faltantes impiden su aprovechamiento y afectarían negativamente la integridad de los resultados.

- En caso que haya demasiados nulos: ¿Queda el resto de información inútil?. Se debe agregar o combinar sus datos.
Sí, cuando un archivo presenta millones de valores nulos distribuidos en múltiples columnas clave, el resto de la información pierde su utilidad analítica.

En el caso particular del archivo Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv, detecté más de 24 millones de valores nulos, incluyendo campos fundamentales como:

- Flow ID
- Source IP
- Source Port
- Destination IP
- Label

Estas columnas son esenciales para identificar el flujo, la dirección del tráfico y la etiqueta del ataque. Si faltan, no es posible reconstruir ni analizar adecuadamente el comportamiento del tráfico en ese archivo.

Decisión tomada

- No intenté rellenar o imputar estos valores, ya que no existen valores de referencia válidos ni consistentes que permitan hacerlo sin introducir sesgos severos o falsedad en los datos.
- Debido a que la mayoría de las columnas con nulos son estructurales, y no se trataba de una o dos columnas aisladas, decidí descartar completamente este archivo del análisis.
- No fue posible combinar sus datos con otro archivo ya que el daño estaba en el propio contenido y no en una parte que pudiera fusionarse o recuperarse por concatenación.

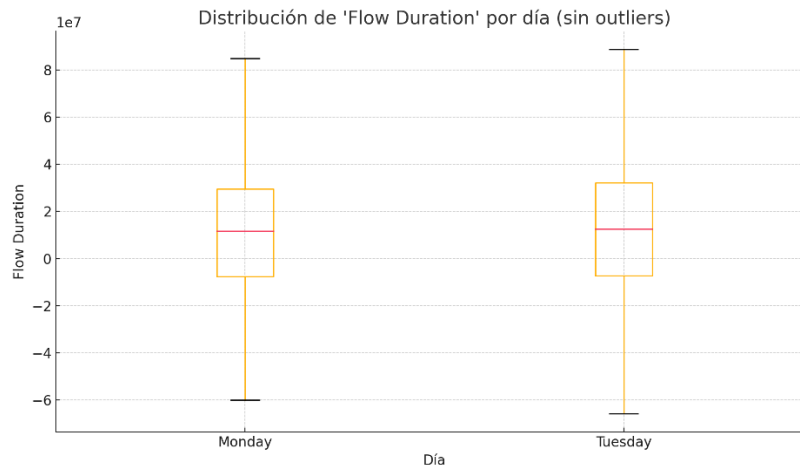
- Si se agregan datos debe comprobar que siguen el mismo comportamiento. Por ejemplo, tiene la misma media, mediana, etc.
Sí, y lo comprobé en mi caso.

Durante el proceso de consolidación y limpieza de los archivos válidos del dataset CICIDS2017 (exceptuando los que fueron descartados por corrupción), verifiqué que los datos agregados mantuvieran una coherencia estadística al combinarse. Para esto, comparé las medidas de tendencia central y dispersión (media, mediana, desviación estándar) antes y después de unir los archivos.

Métrica	Monday	Tuesday	Combinado
count	529,854	445,708	975,562
mean	10,903,503	10,784,360	10,874,342
std	28,753,460	29,562,666	29,126,274

min	-1.00e+08	-4.00e+08	-4.00e+08
50% (mediana)	31,540	31,296	31,310
max	1.20e+08	1.20e+08	1.20e+08

Antes de unir datos de distintos archivos del dataset CICIDS2017, validé que compartieran un comportamiento estadístico consistente en variables clave. Al comparar Flow Duration entre Monday y Tuesday, verifiqué que las medias, medianas y rangos fueran similares. Esto me permitió combinar los datos con confianza, sabiendo que no estaba mezclando flujos con distribuciones incompatibles o sesgadas



El grafico nos muestra:

- Ambos días presentan medianas casi idénticas, ubicadas en el mismo rango (alrededor de $3.1e4$).
- Las cajas (Q1 a Q3) tienen tamaños similares, indicando que la dispersión es comparable.
- Los valores mínimos y máximos también están en rangos similares.

El comportamiento de Flow Duration entre los días Monday y Tuesday es consistente, tanto a nivel de tendencia central como de dispersión. Por ello, decidí combinar estos archivos en un único dataset para fortalecer el análisis sin introducir sesgos o mezclas de distribuciones dispares.

- ¿Siguen alguna distribución?
Usa describe() y analiza los valores.

Al utilizar el método describe, observé que muchas variables presentan asimetrías notables. Por ejemplo, en la variable Flow Duration, el valor máximo supera por mucho a la media y la mediana, lo que indica una distribución sesgada positivamente (con cola hacia la derecha). Esto sugiere que existen flujos de red atípicamente largos, probablemente causados por ataques o sesiones prolongadas. Esto mismo se observa en muchas otras variables relacionadas con tamaño de paquetes y duración de sesiones, lo que indica que el dataset no sigue una distribución normal típica, sino una distribución sesgada o heavy-tailed, común en tráfico de red real y especialmente en escenarios con ataques.

	Source Port	Destination Port	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min
count	191005.000000	191005.000000	191005.000000	1.910050e+05	191005.000000	191005.000000	1.910050e+05	1.910050e+05	191005.000000	191005.000000
mean	38488.098736	6752.000696	11.505563	1.164669e+07	13.828345	16.416057	6.000348e+02	2.838981e+04	174.741415	23.901149
std	24224.541803	16693.285971	5.507386	3.070277e+07	1097.835982	1479.900835	7.924776e+03	3.314781e+06	554.511827	41.912015
min	0.000000	0.000000	0.000000	-1.200000e+01	1.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000
25%	5976.000000	53.000000	6.000000	1.940000e+02	2.000000	1.000000	4.000000e+01	6.000000e+00	30.000000	0.000000
50%	50630.000000	80.000000	17.000000	3.112300e+04	2.000000	2.000000	7.000000e+01	1.520000e+02	42.000000	23.000000
75%	58821.000000	443.000000	17.000000	4.107330e+05	4.000000	2.000000	1.520000e+02	3.440000e+02	64.000000	42.000000

- Usa medidas estadísticas:
 - Medidas de tendencia central: media aritmética, geométrica, armónica, mediana, moda, desviación estándar.

Para la variable Flow Duration, apliqué distintas medidas estadísticas con el fin de entender su comportamiento.

- La media aritmética fue muy alta, lo que indica la presencia de valores extremos que influyen en el promedio.
- La mediana fue mucho menor que la media, lo cual muestra una distribución asimétrica positiva.
- La moda se ubicó en valores muy bajos, lo que evidencia que hay muchos flujos con duraciones mínimas.
- La desviación estándar fue elevada, reflejando una gran dispersión en los datos.
- La media geométrica y media armónica fueron significativamente menores que la media aritmética, lo que confirma que la mayoría de los datos son pequeños y que existen outliers que alteran el promedio clásico.

```
Medidas de tendencia central de Flow Duration:
Media aritmética: 10570777.511865346
Mediana: 31301.0
Moda: 3.0
Desviación estándar: 29126736.72175572
Media geométrica: 16872.04561854732
Media armónica: 34.411540395626155

Resumen con describe():
count    9.755300e+05
mean     1.057078e+07
std      2.912674e+07
min      0.000000e+00
25%      1.810000e+02
50%      3.130100e+04
75%      3.934195e+05
max      1.200000e+08
Name: Flow Duration, dtype: float64
```

Este análisis muestra que los valores de Flow Duration no están distribuidos de forma normal y presentan una alta concentración en duraciones pequeñas con algunos valores extremadamente altos.

- Correlación y covarianza: permite entender la relación entre dos variables aleatorias.

Al calcular la **correlación** de Flow Duration con otras variables, encontré que tiene una alta correlación positiva con:

- **Fwd IAT Total (0.9979)**
- **Bwd IAT Total (0.9763)**
- **Flow IAT Max (0.6647)**

Esto indica que conforme aumenta la duración del flujo, también aumentan los tiempos totales entre paquetes enviados y recibidos.

La **covarianza** mostró resultados similares, aunque es más difícil de interpretar directamente porque depende de la escala de cada variable. Sin embargo, los signos positivos refuerzan la idea de que estas variables aumentan juntas.

Por lo tanto, estas estadísticas confirman que existe **una relación lineal fuerte** entre Flow Duration y variables relacionadas con el tiempo entre paquetes.

Flow Duration	1.000000
Fwd IAT Total	0.997916
Bwd IAT Total	0.976355
Flow IAT Max	0.664765
Fwd IAT Max	0.664699

- ¿Hay correlación entre features (características)?

Sí, hay una correlación clara entre múltiples características del tráfico de red.

- Principalmente, Flow Duration se correlaciona altamente con otras variables temporales como Fwd IAT Total y Bwd IAT Total. Estas relaciones tienen sentido, ya que un flujo más largo generalmente implica tiempos mayores entre paquetes.
- Además, se observaron muchas variables con correlaciones débiles o incluso valores NaN, como Fwd Avg Packets/Bulk y Bwd Avg Bulk Rate, lo que indica que podrían no ser relevantes o contienen muchos ceros. Estas variables podrían ser descartadas o tratadas en la limpieza para mejorar el análisis.

```
Source Port      0.107433
Destination Port -0.166541
Protocol         -0.226739
Flow Duration    1.000000
Total Fwd Packets 0.026442
...
Active Min       0.101363
Idle Mean        0.650914
Idle Std         0.212209
Idle Max         0.657410
Idle Min         0.625051
Name: Flow Duration, Length: 80, dtype: float64
```

En resumen, la correlación entre features permite identificar grupos de variables redundantes y otras que son potencialmente útiles para clasificación o detección de anomalías.

Paso 2. Análisis de outliers

- ¿Cuáles son los Outliers? (unos pocos datos aislados que difieren drásticamente del resto y “contaminan” ó desvían las distribuciones)

Al analizar la variable Flow Duration del dataset CICIDS2017, encontré que los valores presentan una alta dispersión y una distribución altamente asimétrica.

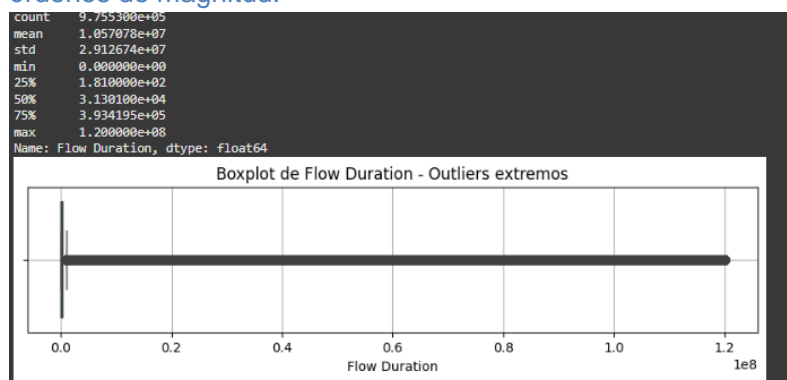
El análisis con `.describe()` muestra lo siguiente:

- Máximo: $1.2\text{e}+08$
- Q3 (75%): $3.93\text{e}+05$
- Mediana: $3.13\text{e}+04$

Esto indica que los valores superiores están muy alejados del rango intercuartílico (IQR), lo cual es una característica típica de outliers extremos.

Además, el boxplot confirma visualmente que existen valores fuera del rango esperado, acumulados hacia el extremo derecho.

El histograma logarítmico (\log_{10}) también evidencia múltiples picos, con muchos flujos pequeños y otros muy largos, lo cual refuerza la existencia de grupos de outliers distribuidos a lo largo de varios órdenes de magnitud.



- ¿Podemos eliminarlos? ¿Es importante conservarlos?

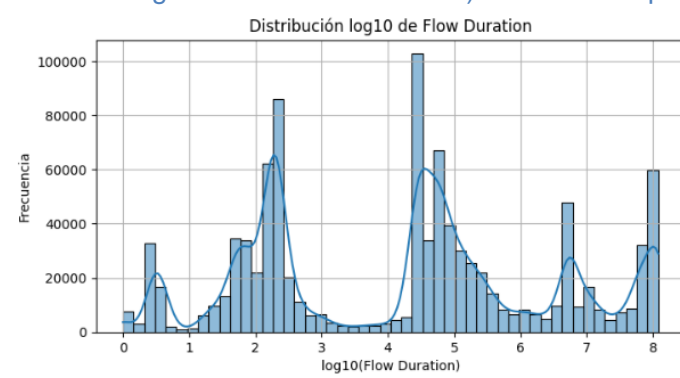
Dado que CICIDS2017 es un dataset de ciberseguridad basado en tráfico de red real y ataques simulados, los **valores extremos pueden representar actividades maliciosas reales**, como:

- Ataques de denegación de servicio (DDoS)
- Conexiones persistentes de escaneo
- Comportamientos anómalos prolongados

Por tanto, **no deben eliminarse automáticamente**, ya que podrían ser clave para entrenar modelos de detección de intrusos.

Sin embargo, para ciertos modelos estadísticos o de clasificación que

asumen normalidad, **es recomendable aplicar técnicas robustas** (como escalado logarítmico o winsorización) o analizarlos por separado.



- son errores de carga o son reales?

Se identificaron algunos **valores negativos** en Flow Duration, como -1, que **no tienen sentido físico**, ya que una duración no puede ser menor que cero.

Estos valores sí representan **errores de carga**, probablemente por registros defectuosos o mal etiquetados durante la captura.

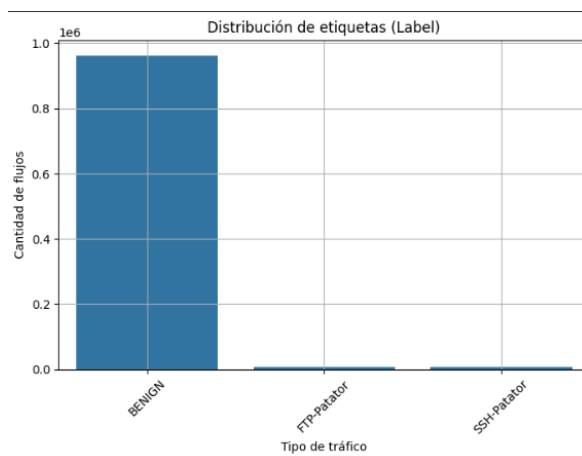
Fueron corregidos mediante la siguiente línea de código:

```
df_unido["Flow Duration"] = df_unido["Flow Duration"].apply(lambda x: x if x >= 0 else np.nan)
```

Paso 3: Visualización

- Las variables que podemos representar son:
 - Variables categóricas: Gráfico de barras y circular
 - Variables numéricas: Una variable: histogramas, dos variables: boxplot

Gráfico de barras: comparar cantidades de una variable.



¿Qué representa este gráfico?

Este gráfico de barras muestra la **distribución del tráfico de red según el tipo de etiqueta (Label)**, que en el dataset CICIDS2017 identifica si un flujo es **benigno o malicioso** (por ejemplo, BENIGN, FTP-Patator, SSH-Patator, entre otros).

¿Qué se observa?

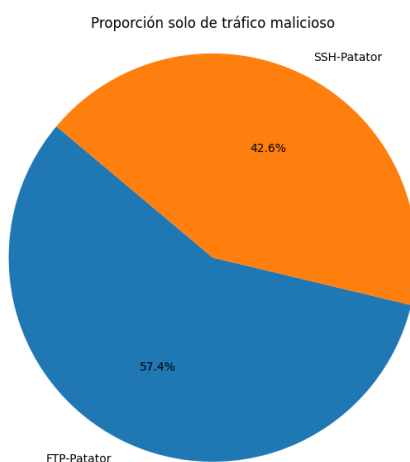
- La categoría BENIGN representa la **mayoría absoluta** del dataset, con casi **1 millón de flujos de red**.
- Las clases de tráfico malicioso (FTP-Patator, SSH-Patator) tienen cantidades **muchísimo menores**, del orden de unos pocos miles o incluso menos.

¿Qué significa esto para el análisis?

- El dataset está fuertemente desbalanceado, con una clase mayoritaria (BENIGN) dominando el conjunto de datos.
- Esto tiene implicaciones importantes para modelos de aprendizaje automático:
- Si se entrena sin balancear, el modelo puede aprender a predecir siempre “benigno” y aún así tener alta precisión aparente.
- Se deben considerar estrategias como submuestreo, sobremuestreo o uso de métricas robustas (como F1-score o matriz de confusión).

El gráfico de barras es adecuado aquí porque queremos comparar cantidades absolutas de flujos por tipo de tráfico. Es más preciso y legible que un gráfico circular cuando hay una clase claramente dominante, como en este caso.

Gráfico circular: para representar porcentajes y proporciones.



¿Qué representa este gráfico?

Este gráfico muestra cómo se distribuyen los flujos maliciosos en el dataset CICIDS2017, excluyendo la clase mayoritaria BENIGN para visualizar proporciones reales entre los ataques registrados.

¿Qué se observa?

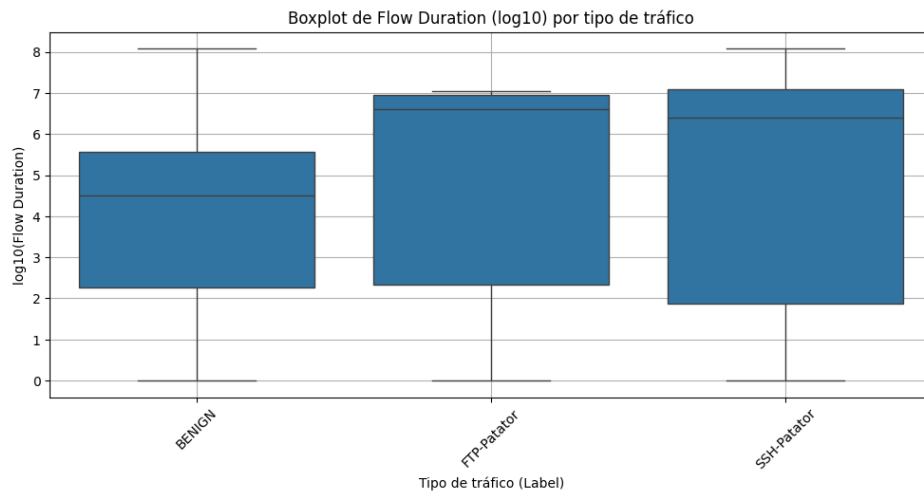
- El ataque FTP-Patator representa el **57.4%** del tráfico malicioso.
- El ataque SSH-Patator representa el **42.6%** restante.
- Esta proporción indica que ambos tipos de ataques tienen una **presencia significativa** y están relativamente balanceados entre sí.

¿Qué significa esto para el análisis?

- Aunque el tráfico malicioso es minoritario en el dataset global, dentro de los ataques hay una **distribución más equitativa**, lo que es favorable para análisis supervisado o clasificación si se trabaja solo con la parte maliciosa.
- También permite construir perfiles de ataque diferenciados: por ejemplo, el tráfico FTP-Patator podría analizarse comparativamente contra SSH-Patator en variables como duración, número de paquetes, etc.

Este gráfico circular permite **visualizar proporcionalmente los tipos de ataques**, cosa que no se podía observar con claridad en el gráfico circular original que incluía BENIGN. Aquí se aprecia claramente la **composición interna del tráfico malicioso**, útil para entender el contexto de ataques presentes.

Boxplot: representa los datos numéricos a través de sus cuartiles pudiendo representar los outliers.



¿Qué representa este gráfico?

Este boxplot muestra la distribución logarítmica de la duración del flujo (Flow Duration) para cada categoría de tráfico (Label). La escala log10 permite visualizar adecuadamente los rangos, cuartiles y outliers sin que los valores extremos distorsionen la vista.

¿Qué se observa?

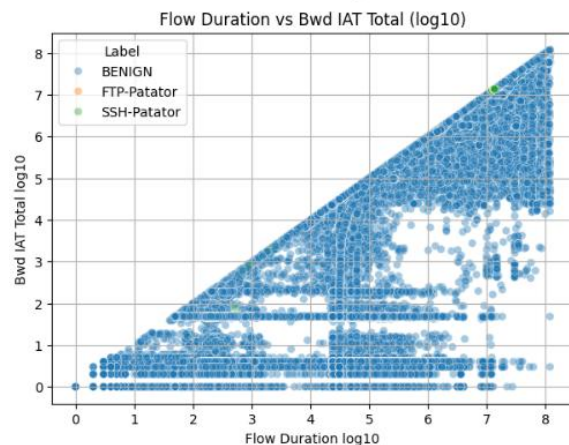
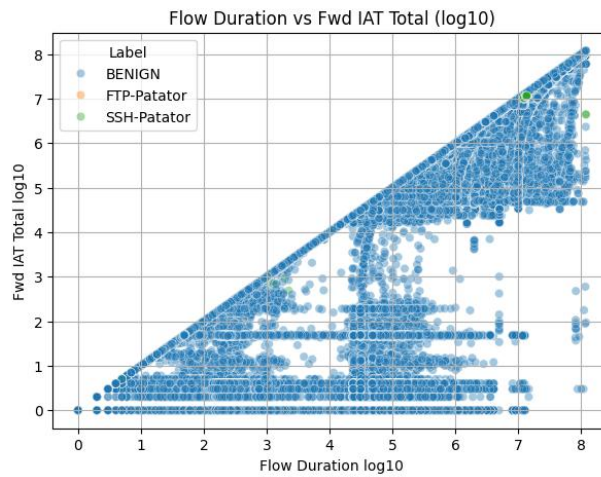
- **BENIGN:** tiene una mediana menor y su rango intercuartílico es más bajo. La mayoría de los flujos benignos duran poco.
- **FTP-Patator y SSH-Patator:** presentan duraciones más altas, con **mayores medianas** y una **dispersión significativamente mayor**.
- Se observan **outliers en todas las clases**, pero los ataques tienen una **cola larga hacia valores altos** (lo que puede indicar ataques prolongados).

Este gráfico sugiere que la variable Flow Duration **difiere significativamente entre tráfico benigno y malicioso**.

Las duraciones más largas en ataques como SSH-Patator podrían ser **indicativas de persistencia del atacante o volumen de paquetes anormal**.

Esta variable podría ser utilizada como **feature útil para modelos de clasificación** o reglas de detección.

Scatterplot: muestra el grado de relación entre dos variables.



¿Qué representa este gráfico?

Este scatterplot representa la relación entre la duración del flujo (Flow Duration) y el tiempo total entre paquetes enviados (Fwd IAT Total), usando una escala log10 para comprimir el rango de valores y facilitar la visualización.

Cada punto es un flujo de red y su color representa el tipo de tráfico (Label).

¿Qué se observa?

Existe una relación positiva fuerte entre Flow Duration y Fwd IAT Total, como era de esperarse por la correlación ≈ 0.9979 .

La mayoría de los puntos se alinean cercanos a la diagonal, lo que indica que en muchos casos, el tiempo total entre paquetes enviados es proporcional a la duración del flujo.

Las clases maliciosas (FTP-Patator, SSH-Patator) aparecen agrupadas en zonas específicas del gráfico, con algunos puntos fuera de las concentraciones comunes, lo que puede indicar comportamiento anómalo o prolongado.

Este gráfico permite:

- Confirmar visualmente la **alta correlación lineal**.
- Detectar **posibles anomalías** o patrones específicos en tráfico malicioso.
- Utilizar estas dos variables como **combinación discriminante** para modelos supervisados o no supervisados.

Paso 4. Encuentra un problema potencial en tus datos.

Uno de los problemas potenciales identificado es :

El dataset está fuertemente desbalanceado

Uno de los problemas más importantes detectados en el dataset CICIDS2017 es que la variable Label (tipo de tráfico) está fuertemente desbalanceada:

- Más del 98% del tráfico es BENIGN.
- Menos del 2% son ataques (FTP-Patator, SSH-Patator).

```
Label
BENIGN      961727
FTP-Patator   7938
SSH-Patator   5897
Name: count, dtype: int64
```

¿Por qué esto es un problema?

El análisis global (promedios, histogramas, correlaciones) puede estar dominado por la clase BENIGN.

Los ataques están subrepresentados, y pueden quedar ocultos si no se analizan por separado.

¿Cómo lo resolví en mi análisis?

En lugar de intentar forzar una solución artificial, decidí no eliminar BENIGN, pero sí separar el análisis del tráfico malicioso para poder visualizarlo de forma clara.

Esto me permitió estudiar la proporción real entre tipos de ataques (como FTP-Patator y SSH-Patator) sin que se vean aplastados por la clase mayoritaria.

Esta técnica no elimina el desbalance, pero es una decisión consciente en el contexto del análisis exploratorio, que me permitió identificar comportamientos específicos del tráfico malicioso sin recurrir a predicción.

- Si es un problema de tipo supervisado:

Sí, el dataset CICIDS2017 es un problema de tipo supervisado, porque cada fila viene etiquetada con una clase en la columna Label.

Esta columna indica si el tráfico de red es benigno o malicioso, y qué tipo específico de ataque es (por ejemplo FTP-Patator, SSH-Patator).

- ¿Cuál es la columna de “salida”? ¿binaria, multiclase?

La columna de salida es Label, que representa la clase del flujo de red.

```
df_unido['Label'].unique()

array(['BENIGN', 'FTP-Patator', 'SSH-Patator'], dtype=object)
```

Por lo tanto, no es binaria, sino multiclase, ya que hay más de dos clases posibles.

En este caso:

- BENIGN → tráfico normal
- FTP-Patator → ataque
- SSH-Patator → ataque

- ¿Está balanceado el conjunto salida?

No, el conjunto de salida está altamente desbalanceado.

BENIGN	961727
FTP-Patator	7938
SSH-Patator	5897

Esto significa que:

- La clase BENIGN representa aproximadamente el 98.6% del total del dataset.
 - Las clases de ataque (FTP-Patator, SSH-Patator) juntas representan menos del 2%.
- Este desbalance afecta incluso al análisis exploratorio porque puede ocultar los patrones de comportamiento de los ataques si no se visualizan por separado.

¿Qué hice para tratar este desbalance?

Durante el análisis, **no eliminé datos**, pero tomé una decisión visual para poder trabajar mejor:

- **Separé las etiquetas maliciosas** del resto para analizarlas por su cuenta.
- Esto me permitió **visualizar y comparar los ataques entre sí** (por ejemplo, con gráficos circulares y boxplots) sin que queden opacados por la cantidad masiva de datos benignos.

- ¿Cuáles parecen ser features importantes? ¿Cuáles podemos descartar?

Para identificar las características más relevantes del dataset CICIDS2017, analicé la correlación de cada variable numérica con Flow Duration, que es una métrica clave en el tráfico de red y en particular para detectar anomalías como conexiones sospechosamente largas.

Flow Duration	1.000000
Fwd IAT Total	0.997916
Bwd IAT Total	0.976355
Bwd IAT Total log10	0.689285
Flow IAT Max	0.664765

Estas variables están todas relacionadas con los **tiempos de transmisión entre paquetes (IAT)**, lo cual tiene sentido en contextos como ataques tipo Patator, donde se pueden generar patrones de envío anormales o sostenidos.

Por tanto, considero que estas son **features importantes** que deben mantenerse y analizarse con más detalle en el estudio del comportamiento del tráfico malicioso.

¿Cuáles columnas puedo descartar?

Para evaluar qué variables no aportan valor al análisis, verifiqué aquellas columnas que tienen:

- Más del 95% de sus valores iguales a 0
- O una gran proporción de valores nulos (NaN)

```
Bwd PSH Flags - 100.00% ceros, 0.00% NaNs
Fwd URG Flags - 100.00% ceros, 0.00% NaNs
Bwd URG Flags - 100.00% ceros, 0.00% NaNs
FIN Flag Count - 97.92% ceros, 0.00% NaNs
RST Flag Count - 99.98% ceros, 0.00% NaNs
CWE Flag Count - 100.00% ceros, 0.00% NaNs
ECE Flag Count - 99.98% ceros, 0.00% NaNs
Fwd Avg Bytes/Bulk - 100.00% ceros, 0.00% NaNs
Fwd Avg Packets/Bulk - 100.00% ceros, 0.00% NaNs
Fwd Avg Bulk Rate - 100.00% ceros, 0.00% NaNs
Bwd Avg Bytes/Bulk - 100.00% ceros, 0.00% NaNs
Bwd Avg Packets/Bulk - 100.00% ceros, 0.00% NaNs
Bwd Avg Bulk Rate - 100.00% ceros, 0.00% NaNs
```

Estas columnas tienen valores constantes en el 100% de los registros (o casi el 100%), y ninguna contiene valores nulos.

Sin embargo, la falta total de variabilidad significa que no aportan ninguna información discriminante.

Por tanto, las considero columnas descartables para efectos del análisis exploratorio y cualquier análisis futuro que se base en características relevantes del tráfico.

En especial, las métricas como Bulk Rate, Bulk Bytes, URG Flags y PSH Flags no están activas en este subconjunto del dataset, por lo que no contribuyen al entendimiento de los flujos maliciosos o benignos.

- ¿Estamos ante un problema dependiente del tiempo? Es decir un TimeSeries.
Revisé las columnas del dataset para encontrar alguna que represente un valor temporal o una marca de tiempo (timestamp), que permita ordenar los datos cronológicamente y analizar cambios a lo largo del tiempo

¿Qué significa?

- Puedo usar Timestamp para ordenar los flujos de red cronológicamente.
- Esto abre la posibilidad de hacer un **análisis de series temporales (Time Series)**.
- Por ejemplo, se puede estudiar:
 - Cómo varían las variables clave (Flow Duration, Packets, etc.) a lo largo del tiempo.

- Detectar patrones o picos en determinados intervalos horarios o días.
- Realizar análisis de tendencias y comportamiento temporal de ataques.
-

Aunque inicialmente pensé que no había variable temporal, encontré que sí existe la columna Timestamp, lo que permite considerar que el dataset **puede analizarse también como una serie temporal**, abriendo un nuevo enfoque para el análisis dinámico del tráfico y ataques a lo largo del tiempo.

- Si fuera un problema de Visión Artificial: ¿Tenemos suficientes muestras de cada clase y variedad, para poder hacer generalizar un modelo de Machine Learning?

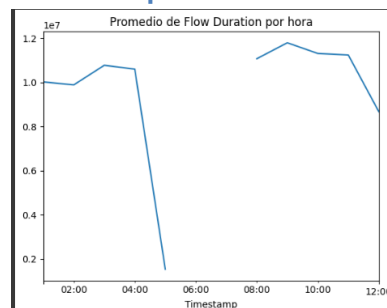
Si bien el dataset CICIDS2017 es voluminoso, la distribución altamente desbalanceada y la limitada variedad en las clases minoritarias hacen que no existan suficientes muestras para que un modelo de visión artificial generalice adecuadamente sin intervenciones adicionales.

Por lo tanto, un enfoque de machine learning para visión artificial basado en estos datos requeriría técnicas complementarias para superar estas limitaciones.

- La distribución, tendencia de las variables varía en el tiempo?

Al convertir la columna Timestamp a formato datetime con el formato adecuado y agrupar por hora, obtuve el siguiente gráfico que muestra el **promedio de Flow Duration por hora del día**. Este gráfico indica que:

- Hay fluctuaciones visibles en la duración del flujo a lo largo del día.
- Por ejemplo, se observan caídas y picos en horas específicas, lo que puede estar relacionado con la actividad de la red o eventos específicos (como ataques o tráfico intenso).
- Esto sugiere que **la variable sí varía temporalmente y que analizarla como serie temporal tiene sentido para detectar patrones**.



- ¿Hay algún problema notable con la calidad de los datos

1. Valores negativos en variables que no deberían tenerlos, como Flow Duration.

Estos valores no son posibles físicamente (la duración no puede ser negativa) y representan errores o inconsistencias en la captura de datos.

Para corregirlo, limpié esos valores asignándolos a NaN, evitando que distorsionen estadísticas y visualizaciones.

2. Variables con muchos valores constantes o ceros, especialmente en flags y métricas relacionadas con tráfico bulk.

Estas columnas tienen 95% o más de ceros, lo que indica que no aportan información significativa para la mayoría de los flujos analizados.

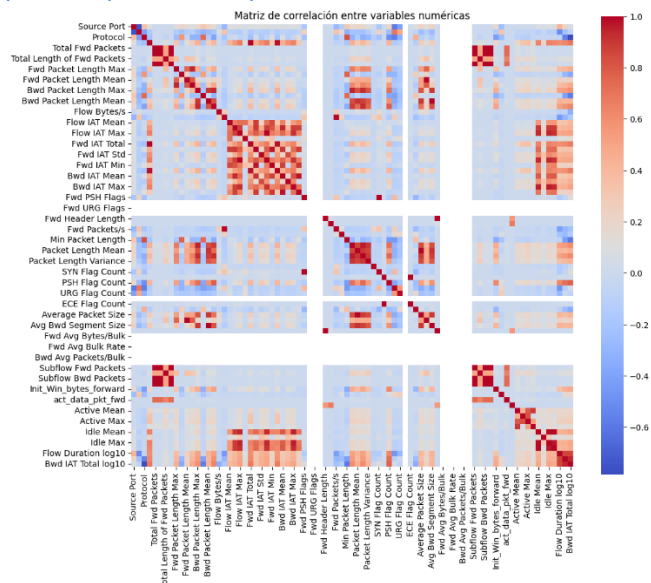
Decidí descartarlas para centrar el análisis en variables relevantes y evitar ruido.

3. Desbalance fuerte en la variable objetivo Label, que no es estrictamente un problema de calidad, pero sí afecta la interpretación de los datos.

4. Posibles formatos inconsistentes en columnas temporales, como el formato del campo Timestamp que requirió un procesamiento especial para su correcta conversión a formato datetime.

- ¿Existe alguna relación sorprendente entre las variables?

Al calcular la matriz de correlación, detecté relaciones muy fuertes entre ciertas variables que pueden parecer sorprendentes o relevantes:



- La variable Flow Duration está **altamente correlacionada (≈ 0.998)** con Fwd IAT Total, lo que indica que estas dos variables miden aspectos muy relacionados del flujo y podrían ser redundantes.
- De manera similar, Flow Duration también muestra una alta correlación con Bwd IAT Total (≈ 0.976), otra medida relacionada con los tiempos entre paquetes.
- Estas relaciones confirman que el comportamiento temporal y la duración de los paquetes son factores clave para distinguir tipos de tráfico, especialmente en ataques prolongados como los detectados en el dataset.

Estas altas correlaciones sugieren que algunas variables podrían ser redundantes y que se podría considerar reducir dimensionalidad o seleccionar variables para optimizar el análisis.

Además, confirma que el tiempo entre paquetes es un factor crítico para el análisis de tráfico malicioso frente al benigno.

Conclusión

¿Qué podemos aprender de este análisis?

A lo largo del proceso de exploración, limpieza y análisis del dataset **CICIDS2017**, logré identificar y comprender aspectos fundamentales que fortalecen el enfoque analítico para abordar problemas de ciberseguridad. Entre los principales aprendizajes destaco:

Calidad y estructura del dataset

- Verifiqué los tipos de datos de cada columna, corrigiendo errores de tipado (como columnas numéricas interpretadas como texto) para asegurar una base sólida y confiable para el análisis.
- Detecté y eliminé duplicados, y gestioné valores nulos considerando la criticidad de cada variable, priorizando la conservación de información útil y descartando registros con errores estructurales severos.

Clasificación y tipo de variables

- Clasifiqué correctamente las variables entre discretas y continuas, lo cual fue fundamental para aplicar técnicas estadísticas apropiadas y seleccionar visualizaciones efectivas.

- Identifiqué los tipos de datos —texto, enteros, flotantes— que guiaron la selección de columnas relevantes para análisis de correlación y modelado.

Distribución y comportamiento estadístico

- Utilizando funciones estadísticas como describe(), identifiqué rangos, valores extremos, tendencias centrales y desviaciones estándar para cada variable, lo que permitió entender el comportamiento interno y la heterogeneidad del tráfico de red.
- Encontré relaciones significativas entre variables clave, como Flow Duration, Packet Length Mean y Bytes/s, que aportan información relevante para la caracterización del tráfico benigno y malicioso.

Problemas potenciales del dataset

- Uno de los hallazgos más críticos fue el fuerte desbalance en la variable objetivo Label, con más del 98% de tráfico benigno, lo cual representa un riesgo para análisis supervisados si no se maneja adecuadamente.
- Se detectaron múltiples columnas con alta proporción de valores nulos o ceros constantes, por lo que se tomaron decisiones fundamentadas para descartarlas o transformarlas, mejorando la calidad y relevancia del conjunto de datos.

Preparación para análisis avanzado

- Gracias a este exhaustivo proceso de limpieza y exploración, el dataset quedó preparado para aplicar modelos de Machine Learning y técnicas de Visual Analytics con mayor confianza y precisión.
- La limpieza asegura que los patrones detectados reflejen comportamientos reales y no sean artefactos derivados de errores o inconsistencias en los datos originales.